Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Računarstvo usluga i analiza podataka

SEMINARSKI RAD

„Predviđanje bolesti na temelju simptoma“

Tomislav Celić

Ante Dragun

Osijek, 2025.

Sadržaj

1. UVOD 1

2. OPIS PROBLEMA 2

2.1. Korišteni podaci 2

2.2. Obrada podataka 4

2.3. Korišteni postupci strojnog učenja 5

3. OPIS PROGRAMSKOG RJEŠENJA 8

3.1. Model strojnog učenja 8

3.2. Način korištenja API-ja 9

3.3. Klijentska aplikacija 12

3.4. Dodatno 14

3.4.1. Izlaganje API-ja za dohvaćanje podataka iz baze podataka 14

3.4.2. Podešavanje hiperparametara modela 16

4. ZAKLJUČAK 19

5. POVEZNICE I LITERATURA 20

1. UVOD

Točna i pravovremena dijagnoza ključna je u liječenju svih bolesti i poboljšavanja zdravstvenih stanja. Klasičnu dijagnozu bolesti utvrđuje liječnik, ali često zbog današnje potrebe za brzim dobivanjem informacija, ljudi koriste Internet za otkrivanje dijagnoza na temelju simptoma bolesti umjesto odlaska liječniku. Iako dijagnosticiranje putem Interneta nije najučinkovitija metoda zbog širokog spektra simptoma kod mnogih bolesti, i dalje pruža dobar inicijalni uvid u zdravstveno stanje te olakšava ljudima razumijevanje mogućih uzroka njihovih simptoma. Modeli strojnog učenja u tom području, zbog svoje sposobnosti analize velikih količina podataka i prepoznavanja obrazaca koji bi eventualno mogli ukazati na određenu bolest, pronalaze svoju svrhu. Primjena takvih modela može poboljšati točnost i pravovremenost dijagnoza, smanjiti opterećenje zdravstvenih sustava, omogućujući liječnicima da usmjere pažnju na složenije kliničke slike. Korištenjem stvarnih medicinskih podataka, modeli strojnog učenja mogu biti dobro istrenirani za prepoznavanje složenih odnosa između simptoma i mogućih dijagnoza. Stvaranje kvalitetnih modela obično predstavlja veliki izazov, zbog toga što bi najbolji modeli bili istrenirani na stvarnim podacima, do kojih je teško doći zbog privatnosti pacijenata. Također je važno napomenuti da, iako takvi modeli mogu ponuditi precizne preporuke, konačna dijagnoza uvijek mora biti u rukama liječnika.

Cilj ovog seminarskog rada je kreirati aplikaciju koja bi na temelju unesenih simptoma od strane korisnika, predvidjela najvjerojatnije bolesti s obzirom na te simptome te ih za daljnje upute usmjerila na službenu stranicu privatnog američkog medicinskog centra *Mayo Clinic*. U seminarskom je radu detaljno opisan problem, korišteni podaci i postupci strojnoga učenja, opisano je kompletno programsko rješenje te je ukratko prikazan rad aplikacije.

2. OPIS PROBLEMA

Iako se većina modela strojnog učenja za dijagnosticiranje bolesti na temelju simptoma ograničava na specifična područja ili čak specifične bolesti, postoje i modeli koji predviđaju širi spektar dijagnoza. Obično takvi modeli pokazuju značajna ograničenja u preciznosti i općenitoj kvaliteti, a primarni se problemi pojavljuju zbog nedostatka personalizacije, loše kvalitete podataka te nepostojeće integracije sa stvarnim medicinskim protokolima.

Nedostatak personalizacije u dijagnozi od strane modela strojnog učenja odnosi se na ignoriranje anamneze i životnog stila korisnika. Modeli za dijagnosticiranje obično se oslanjaju na opće statističke modele i fiksna pravila te zbog toga obično budu previše generička ili ne odgovaraju specifičnim korisnicima. Kao što je prethodno spomenuto, loša kvaliteta podataka koji se uzimaju za treniranje modela rezultiraju lošijim performansama modela, a kao najveći problem u stvaranju modela koji se koriste u medicini su ograničenja u pristupu stvarnim medicinskim podacima. Integracija stvarnih medicinskih protokola kao što su daljnje medicinske upute i savjeti obično se ne povezuju s modelima strojnog učenja, a korisnicima to otežava prelazak s digitalne procjene na stručnu medicinsku intervenciju. Aplikacija koja je izrađena u sklopu ovog seminarskog rada - *Illness Predictor*, za svaku prikazanu bolest omogućuje pristup poveznici na službenu stranicu *Mayo Clinic*a, koji detaljnije opisuje bolesti koje model predvidi na temelju simptoma koji su uneseni.

Primjer jednog široko korištenog alata koji predviđa bolesti na temelju unesenih simptoma je *WebMD Symptom Checker*. Alat putem vrlo intuitivnog sučelja omogućuje korisnicima unos simptoma i nekih općih podataka kao što su dob i spol, a nakon završetka unosa, aplikacija omogućuje ispis liste nevjerojatnijih bolesti s obzirom na simptome, s ispisom ozbiljnosti stanja, rijetkosti bolesti te ostalim simptomima koji se obično pojavljuju uz tu bolest, kao i metodama liječenja.

2.1. Korišteni podaci

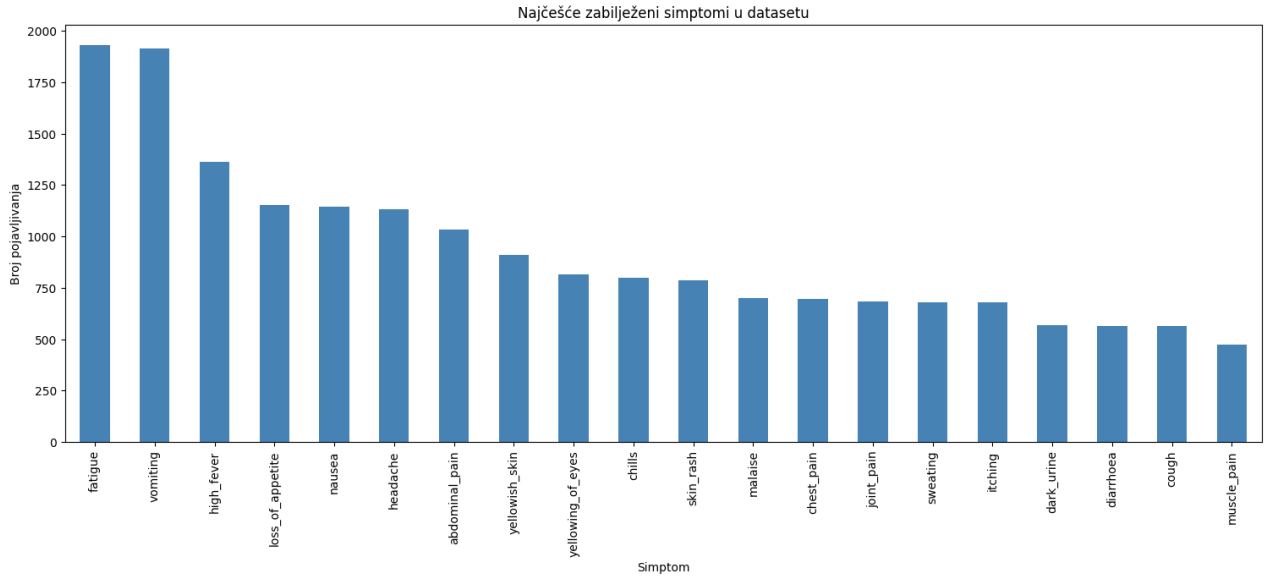
Za izradu modela, potreban je skup podataka na kojemu će se model trenirati za prepoznavanje bolesti. Skup podataka koji se koristio za treniranje modela sadrži 4962 uzorka, pri čemu su simptomi, kojih ima 132, korišteni kao značajke, dok je dijagnoza bolesti korištena kao oznaka. S obzirom na broj simptoma (132), broj bolesti i njihovu učestalost u stvarnosti, ovaj skup podataka može se smatrati relativno malim u usporedbi s medicinskim bazama podataka koje

često sadrže desetke tisuća uzoraka. Svaka značajka (simptom) binarno je kodirana, gdje 1 označava prisutnost, a 0 odsutnost određenog simptoma kod korisnika. Na slici 1. prikazan je ispis nasumičnih pet uzoraka iz skupa podataka, s ispisom indeksa, 4 simptoma (*itching, skin_rash, nodal_skin_eruptions, yellow_crust_ooze*) i prognozom bolesti.

	itching	skin_rash	nodal_skin_eruptions	...	yellow_crust_ooze	prognosis
0	0	0	0	...	0	Hypothyroidism
1	0	0	0	...	0	Varicose veins
2	1	1	0	...	0	Chicken pox
3	1	1	0	...	0	Fungal infection
4	0	0	0	...	0	Hepatitis C

Slika 1. *Prikaz nasumičnih pet uzoraka iz početnog skupa podataka*

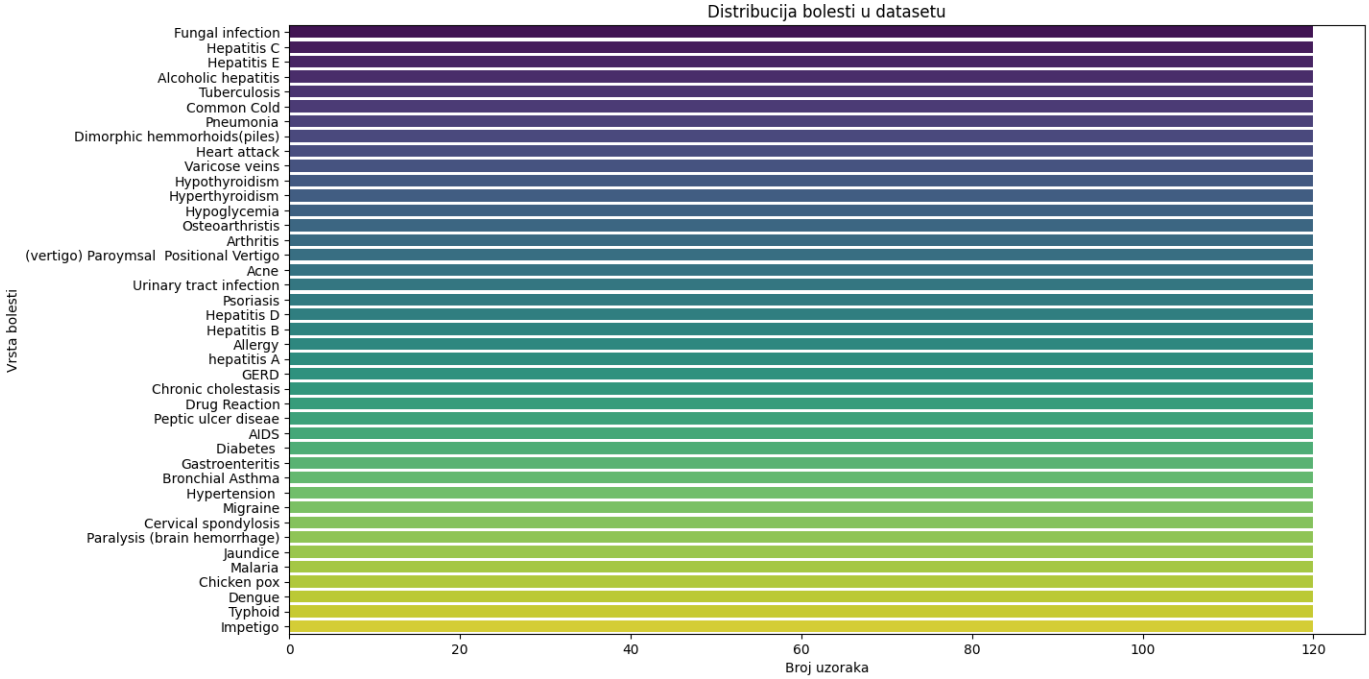
Na slici 2. prikazan je stupčasti dijagram, koji prikazuje najučestalije simptome. Iz dijagrama je vidljivo da je najučestaliji simptom u skupu podataka umor, kojeg slijede povraćanje te povišena tjelesna temperatura, koji se u većini populacije nerijetko pojavljuju.



Slika 2. *Stupčasti dijagram najčešće zabilježenih simptoma u skupu podataka*

Čest problem kod skupova podataka koji se rabe u treniranju modela strojnog učenja je nebalansiranost podataka, odnosno pojavu u kojoj određene klase imaju znatno više primjera od drugih. Ova nebalansiranost može negativno utjecati na performanse modela, jer algoritmi učenja često favoriziraju većinske klase, dok manje zastupljene klase ostaju zanemarene. Slika

3. prikazuje distribuciju bolesti u skupu podataka. Iz slike je vidljivo da se sve klase pojavljuju jednak broj puta u uzorcima, što je znak dobre distribucije bolesti, pa model ne bi trebao biti pristran prema učestalijim bolestima, ali također može predstavljati i izazov u treniranju modela, jer u stvarnosti liječnici kao parametar u određivanju dijagnoze koriste i vjerojatnost pojavljivanja potencijalnih bolesti, a modeli se oslanjaju na statističke podatke iz skupa.



Slika 3. *Distribucija bolesti u skupu podataka*

2.2. Obrada podataka

S obzirom na to da skup podataka sadrži velik broj stupaca, postoji opasnost da bi se model pri treniranju mogao previše oslanjati na trening podatke, pri čemu ne bi pamtio uzorke pojavljivanja simptoma, nego bi naučio specifične kombinacije simptoma, što rezultira *overfittingom*. U takvom slučaju, model bi mogao imati visoku točnost na podacima za treniranje, ali lošu sposobnost generalizacije na nove, neviđene podatke. Kako bi se smanjio rizik od prenaučenosti, korišten je *Chi-squared* test za odabir 80 najznačajnijih značajki. Ova metoda procjenjuje povezanost svake značajke s oznakom klase (bolesti) te omogućuje uklanjanje manje relevantnih simptoma, čime se smanjuje dimenzionalnost skupa podataka i poboljšava učinkovitost modela. Iz početnog skupa podataka uklonjen je stupac koji je sadržavao isključivo null vrijednosti, a koji je nastao zbog nepravilnosti u izvornom CSV

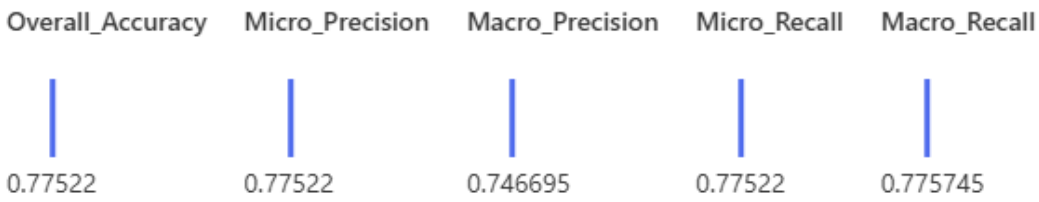
formatu. Konkretno, u datoteci je zaglavlje sadržavalo dodatni nepotreban zarez na kraju, što je rezultiralo stvaranjem praznog stupca prilikom učitavanja podataka. Budući da stupac nije sadržavao nikakve informacije, uklonjen je iz početnog skupa. S obzirom na to da su vrijednosti značajke binarne vrijednosti, normalizacija podataka nije bila potrebna.

2.3. Korišteni postupci strojnog učenja

Podaci su podijeljeni u skup za testiranje i skup za treniranje, gdje skup za treniranje sadrži 70% ukupnog broja uzoraka iz očišćenog podatkovnog skupa, a preostalih 30% nalazi se u skupu za testiranje, a takav je omjer napravljen na temelju rizika od prenaučenosti modela. S obzirom na to da je cilj projekta predvidjeti bolest na temelju unesenih simptoma i da postoji veći skup bolesti, korišteni su modeli višeklasne klasifikacije. Kako bi prognoza bila što uspješnija, trenirano je više modela, a onaj s najvećom ukupnom točnošću odabran je za daljnje korištenje.

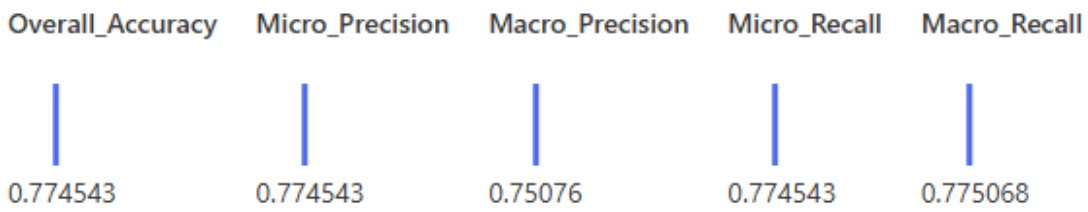
Modeli koji su korišteni su: *Multiclass Logistic Regression, Multiclass Boosted Decision Tree, Multiclass Neural Network, Multiclass Decision Forest te Two Class Support Vector Machine u kombinaciji s One vs All Multiclass*.

Prema [1], *Multiclass Logistic Regression* je klasifikacijska metoda koja generalizira logističku regresiju na višeklasnim problemima, odnosno na one s dva ili više mogućih diskretnih ishoda. Obična logistička regresija koristi se za predviđanje vjerojatnosti događaja uz pomoć prilagodbe podataka logističkoj krivulji, a zavisna varijabla je binarna. Vrijednosti metrika za model *Multiclass Logistic Regression* prikazane su na slici 4.



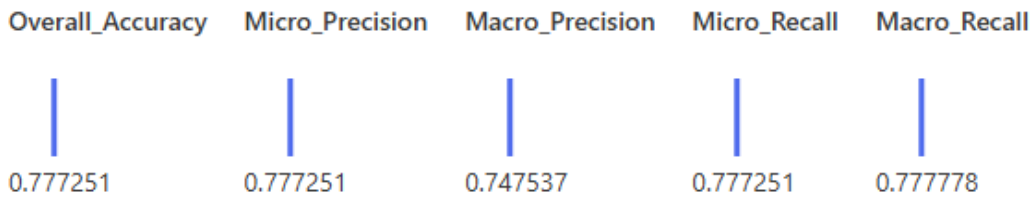
Slika 4. *Vrijednosti metrika za Multiclass Logistic Regression*

Prema [2], *Multiclass Boosted Decision Tree* metoda je učenja koja kombinira više jednostavnijih modela odlučujućih stabala, u kojoj svako stablo ispravlja pogreške prethodnih stabala. Predviđanja se temelje na ansamblu stabala zajedno. Vrijednosti metrika za model *Multiclass Boosted Decision Tree* prikazane su na slici 5.



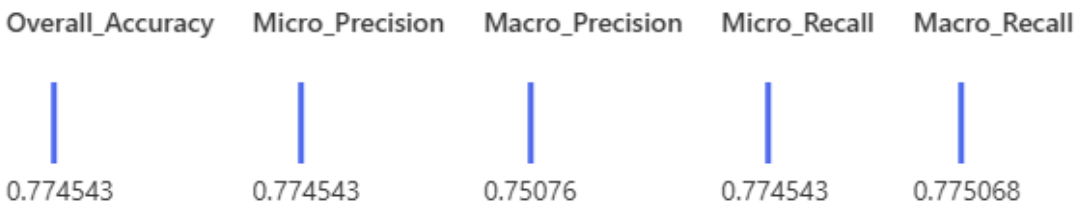
Slika 5. *Vrijednosti metrika za Multiclass Boosted Decision Tree*

Prema [3], *Multiclass Neural Network* predstavlja model dizajniran za prepoznavanje složenih obrazaca u podacima. U kontekstu višeklasne klasifikacije, neuronska mreža ima izlazni sloj s više neurona, gdje svaki neuron predstavlja jednu klasu. Korištenjem aktivacijskih funkcija, neuronska mreža generira vjerojatnosti pripadnosti svakoj klasi, omogućujući klasifikaciju ulaznih podataka u jednu od više kategorija. Vrijednosti metrika za *Multiclass Neural Network* prikazane su na slici 6.



Slika 6. *Vrijednosti metrika za Multiclass Neural Network*

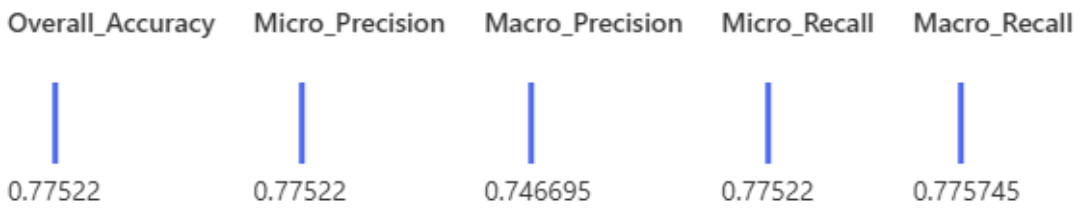
Prema [4], *Multiclass Decision Forest* metoda je učenja za klasifikaciju, a radi tako da gradi više stabala odlučivanja i zatim „glasa“ o najpopularnijoj izlaznoj klasi. Često se koriste za prikazivane nelinearnih granica odlučivanja, a posebno je korisna zbog otpornosti na pojavu šumova u značajkama. Vrijednosti metrika za *Multiclass Decision Forest* prikazane su na slici 7.



Slika 7. *Vrijednosti metrika za Multiclass Decision Forest*

Prema [5], *Support Vector Machine* model je nadziranog učenja koji u procesu učenja analizira ulazne podatke i prepoznaje obrasce u višedimenzionalnom prostoru značajki. Svi su ulazni

primjeri predstavljeni kao točka u prostoru i mapirani su u izlazne kategorije, tako da su kategorije podijeljene što većim razmakom. *One vs All model*, prema [6], omogućuje nekim binarnim klasifikacijskim algoritmima (kao što je SVM u ovom slučaju) prilagodbu za zadatke višeklasne klasifikacije. U OvA, stvara se binarni model za svaku od izlaznih klasa te se potom procjenjuje svaki od binarnih modela za pojedinačne klase u odnosu na sve druge, kao da je problem binarne klasifikacije. Tada se rezultati tog skupa binarnih modela spajaju u jedan model koji predviđa sve klase. Na slici 8. prikazane su metrike *Support Vector Machine*-a u kombinaciji s *One vs All*.



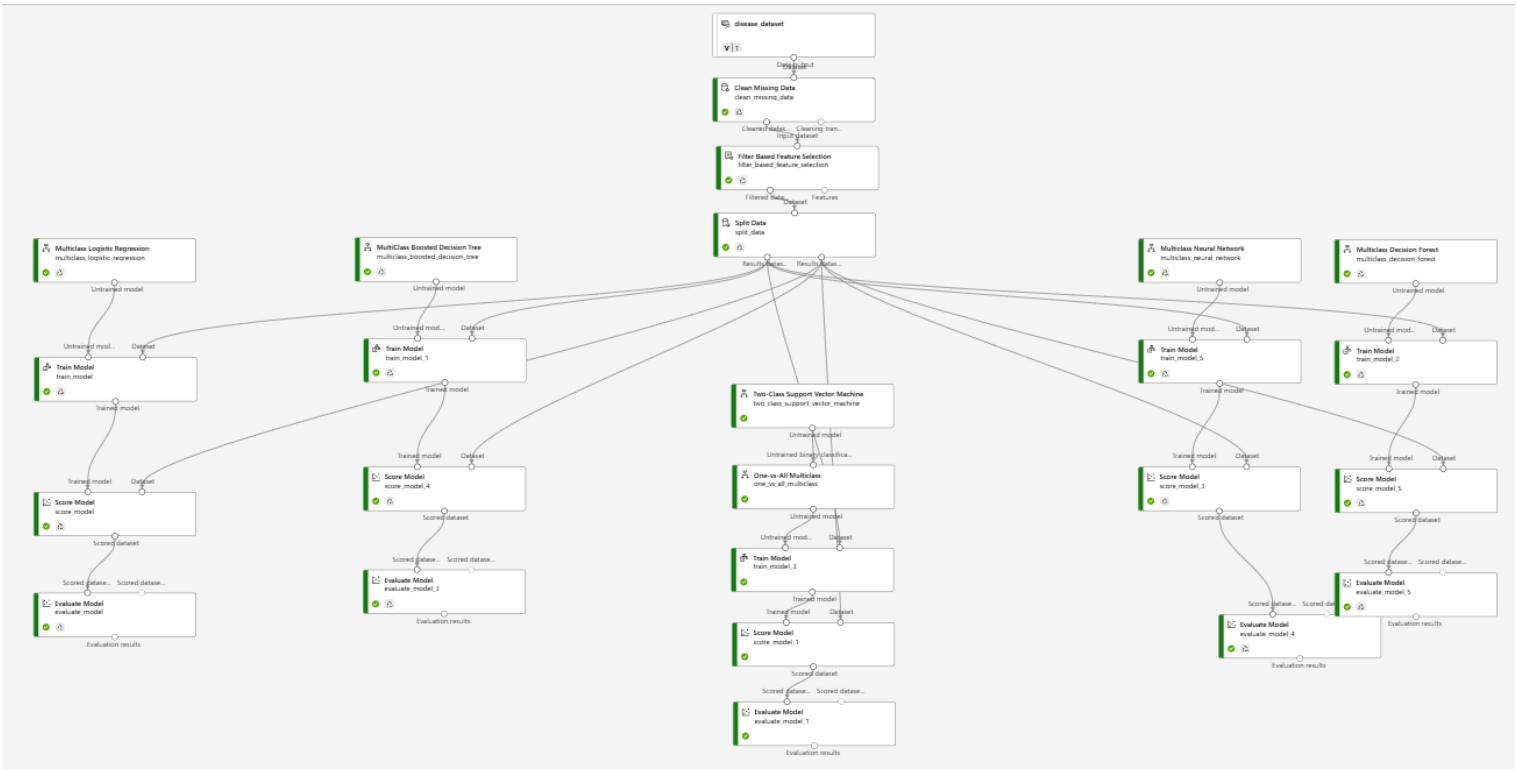
Slika 8. *Vrijednosti metrika za Support Vector Machine (OvA)*

Nakon usporedbe vrijednosti metrika svih modela, prvenstveno ukupne točnosti (engl. *Overall accuracy*), odabran je najpouzdaniji model – *Multiclass Neural Network*, čija je ukupna točnost 77.73%.

3. OPIS PROGRAMSKOG RJEŠENJA

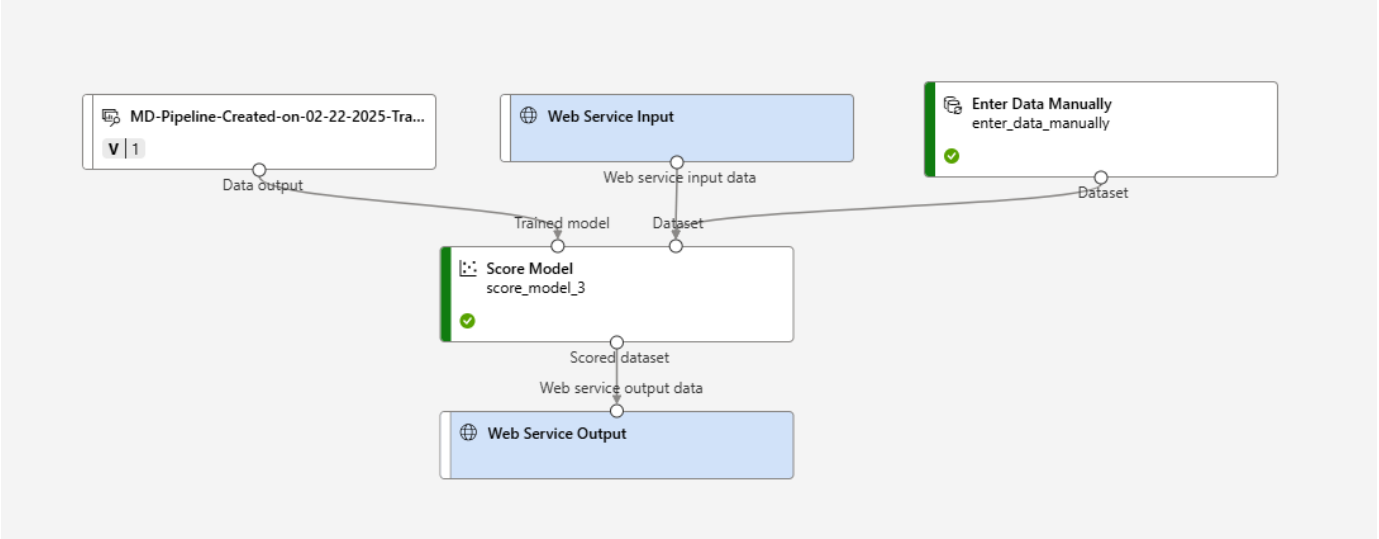
3.1. Model strojnog učenja

Model strojnog učenja izrađen je unutar *Azure Machine Learning Studio*, koristeći *Designer*. Na slici 9. prikazan je cjevovod, kojem je izvorna točka podatkovni skup koji je prethodno opisan. S obzirom na to da je u podatkovnom skupu zbog pogreške u datoteci kreiran nepotreban stupac, blok *Clean Missing Data* ga je obrisao te proslijedio u blok za odabir značajki. Odabrano je najboljih 80 značajki na temelju *Chi-squared* statističkog testa. Podatkovni skup je zatim razdvojen na podatke za trening i test u omjeru 70:30 te se nakon toga izvršava treniranje svih prethodno opisanih modela, čije su metrike kvalitete evaluirane u bloku *Evaluate Model.*



Slika 9. *Prikaz cjevovoda za obradu podataka i treniranje modela*

Nakon što su svi modeli istrenirani te je odabran najbolji, stvara se novi cjevovod za predviđanje u stvarnom vremenu (engl. *Real-time inference pipeline*), koji sadrži *Web Service* ulaz i izlaz koji će se koristiti u konačnoj aplikaciji, a cjevovod je prikazan prikazan na slici 10.

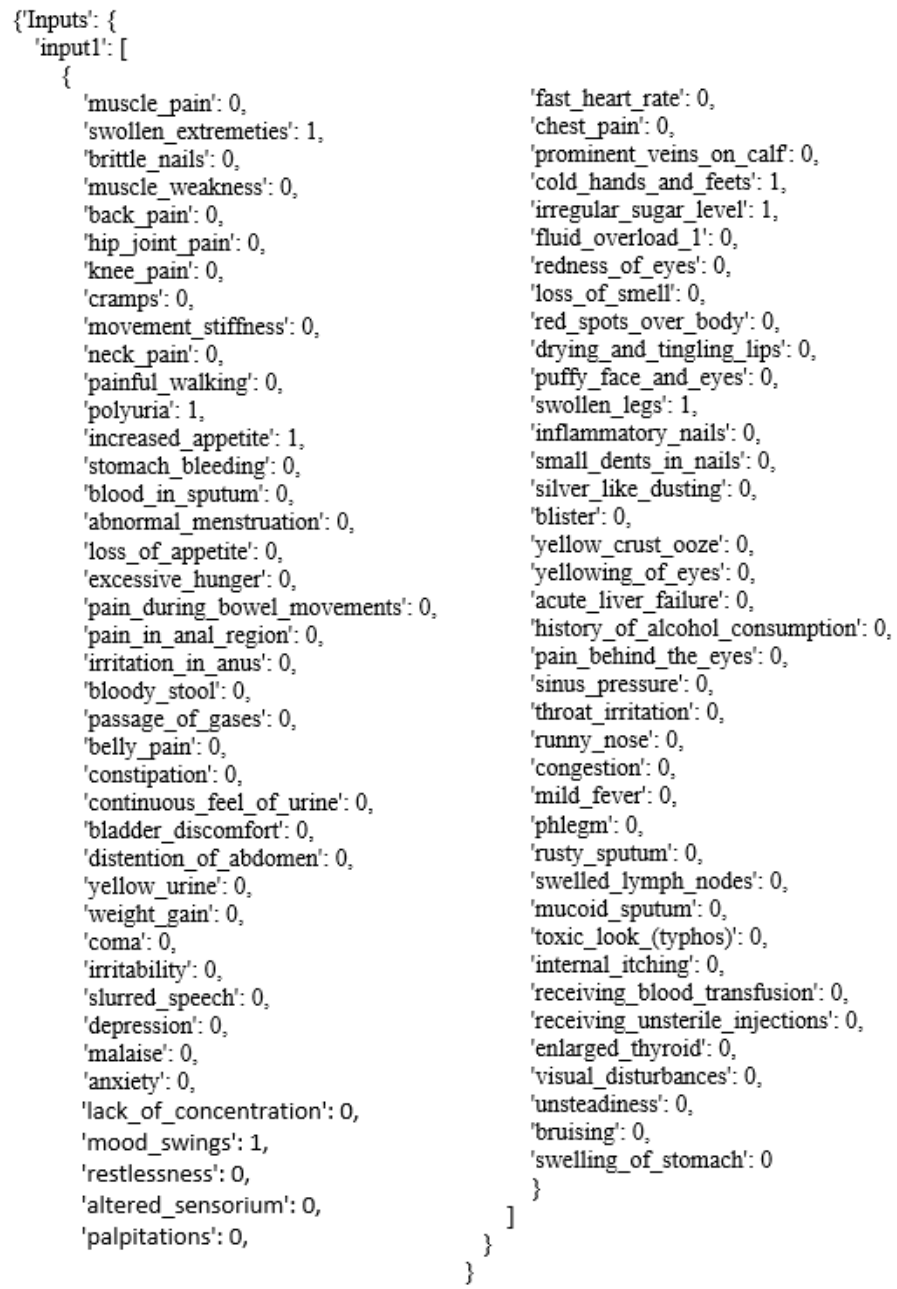


Slika 10. *Prikaz cjevovoda za predviđanje u stvarnom vremenu*

3.2. Način korištenja API-ja

Nakon što je stvoren *Real-time inference pipeline*, kako bi vanjske aplikacije mogle koristiti istrenirani model, potrebno je cjevovod pustiti u produkciju (engl. *deploy*). Kada Azure kreira sve potrebne resurse i pokrene ih, stanje *deploymenta* postaje *healthy* te krajnja točka (engl. *endpoint*) postane dostupna putem URL-a koji omogućava aplikacijama slanje HTTP zahtjeva i dobivanje predikcije modela. Osim URL-a, vanjske aplikacije moraju imati i potreban API ključ, koji služi kao autorizacija, pa ako vanjska aplikacija nema ispravan API ključ, API će odbiti zahtjev.

Aplikacija u sklopu seminarskog rada – *Illness Predictor*, prikuplja simptome kroz web formu i formatira ih u JSON oblik kako bi ih poslala modelu na predviđanje. Na slici 11. nalazi se primjer JSON formatiranog zapisa simptoma koji se prosljeđuje modelu te predstavlja zahtjev (engl. *request*).



Slika 11. *Primjer zahtjeva*

Kada model primi podatke, on ih obradi te odgovara s generiranim predikcijama za svaku klasu i ispiše njihove odgovarajuće vjerojatnosti, a na kraju ispiše i naziv klase s najvećom vjerojatnosti. Odgovor (engl. *response*) koji je poslan vanjskoj aplikaciji također je u JSON formatu kao i zahtjev, a odgovor na obrađeni zahtjev sa slike 11, ispisan je na slici 12.

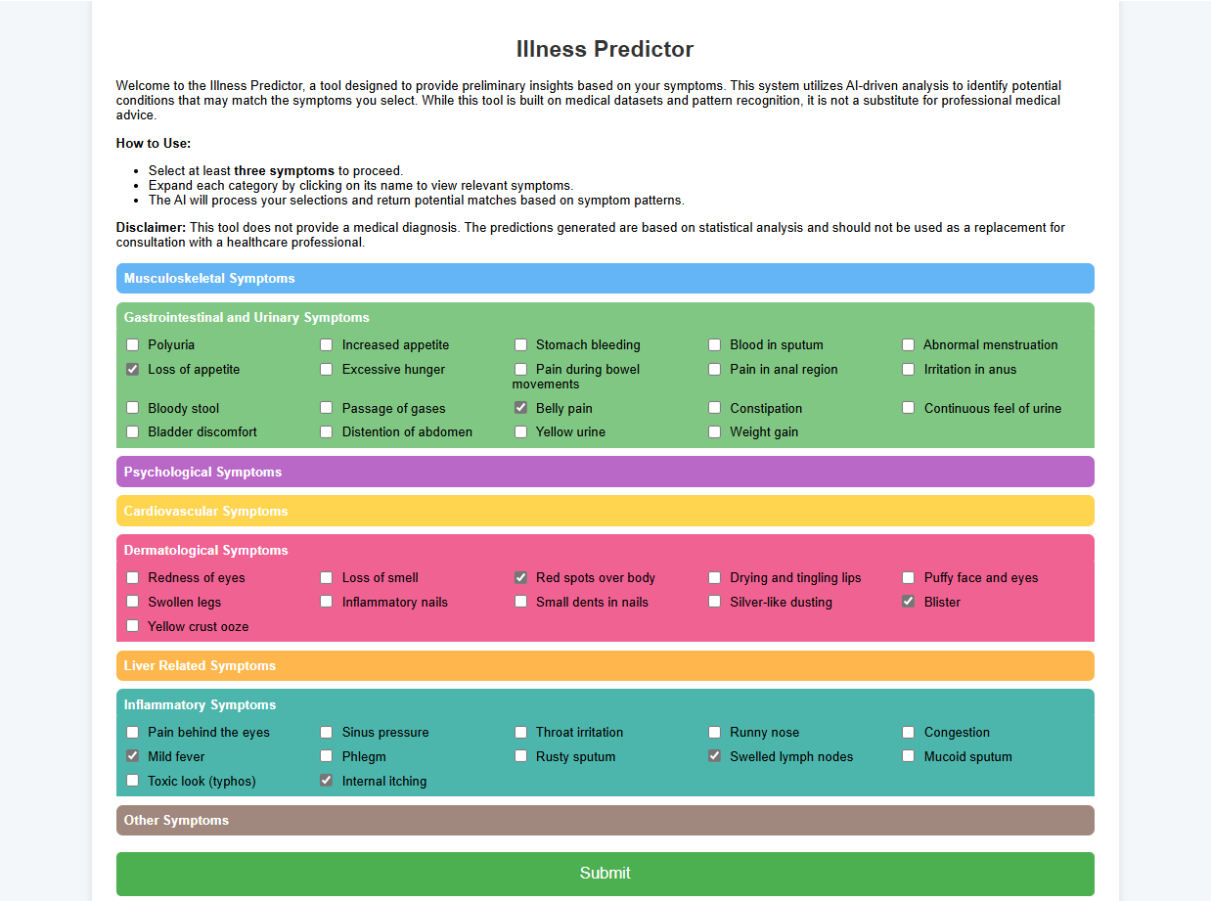
```
"Results": {
  "WebServiceOutput0": [
    {
      "Scored Probabilities_(vertigo) Paroymsal  Positional Vertigo": 0.0038378985209207503,
      "Scored Probabilities_AIDS": 0.015624477524533801,
      "Scored Probabilities_Acne": 0.01589465880177785,
      "Scored Probabilities_Alcoholic hepatitis": 0.004409597781694603,
      "Scored Probabilities_Allergy": 0.013705778604677369,
      "Scored Probabilities_Arthritis": 0.0011169205529138679,
      "Scored Probabilities_Bronchial Asthma": 0.0028977258322267674,
      "Scored Probabilities_Cervical spondylosis": 0.0004633113314623698,
      "Scored Probabilities_Chicken pox": 0.0014574473224899223,
      "Scored Probabilities_Chronic cholestasis": 0.0022113485228010153,
      "Scored Probabilities_Common Cold": 0.0010940761101404523,
      "Scored Probabilities_Dengue": 0.000216587753014373,
      "Scored Probabilities_Diabetes ": 0.7089165707587334,
      "Scored Probabilities_Dimorphic hemmorhoids(piles)": 0.0037195989789533847,
      "Scored Probabilities_Drug Reaction": 0.013790109200763786,
      "Scored Probabilities_Fungal infection": 0.02325688133405274,
      "Scored Probabilities_GERD": 0.01023862014396947,
      "Scored Probabilities_Gastroenteritis": 0.020719837723371177,
      "Scored Probabilities_Heart attack": 0.022985896609785776,
      "Scored Probabilities_Hepatitis B": 0.00031122278135408164,
      "Scored Probabilities_Hepatitis C": 0.0012603120390024356,
      "Scored Probabilities_Hepatitis D": 0.0009722224108182918,
      "Scored Probabilities_Hepatitis E": 0.003385957032554732,
      "Scored Probabilities_Hypertension ": 0.006369797595858252,
      "Scored Probabilities_Hyperthyroidism": 0.002339097390243504,
      "Scored Probabilities_Hypoglycemia": 0.0018734174792483897,
      "Scored Probabilities_Hypothyroidism": 0.018611266346099418,
      "Scored Probabilities_Impetigo": 0.017445239306839642,
      "Scored Probabilities_Jaundice": 0.009527699087294781,
      "Scored Probabilities_Malaria": 0.0003499284268815625,
      "Scored Probabilities_Migraine": 0.0009092268567889605,
      "Scored Probabilities_Osteoarthristis": 0.0005226760916305394,
      "Scored Probabilities_Paralysis (brain hemorrhage)": 0.007527421826297883,
      "Scored Probabilities_Peptic ulcer diseae": 0.0077648994034551725,
      "Scored Probabilities_Pneumonia": 0.0004294498588158253,
      "Scored Probabilities_Psoriasis": 0.007724846716517917,
      "Scored Probabilities_Tuberculosis": 0.0002146569168570567 5,
      "Scored Probabilities_Typhoid": 0.0005037484124637869,
      "Scored Probabilities_Urinary tract infection": 0.001588334674031914,
      "Scored Probabilities_Varicose veins": 0.04359177546513877,
      "Scored Probabilities_hepatitis A": 0.00021946047352422484,
      "Scored Labels": "Diabetes "
    }
  ]
}
```


Slika 12. *Primjer odgovora*

3.3. Klijentska aplikacija

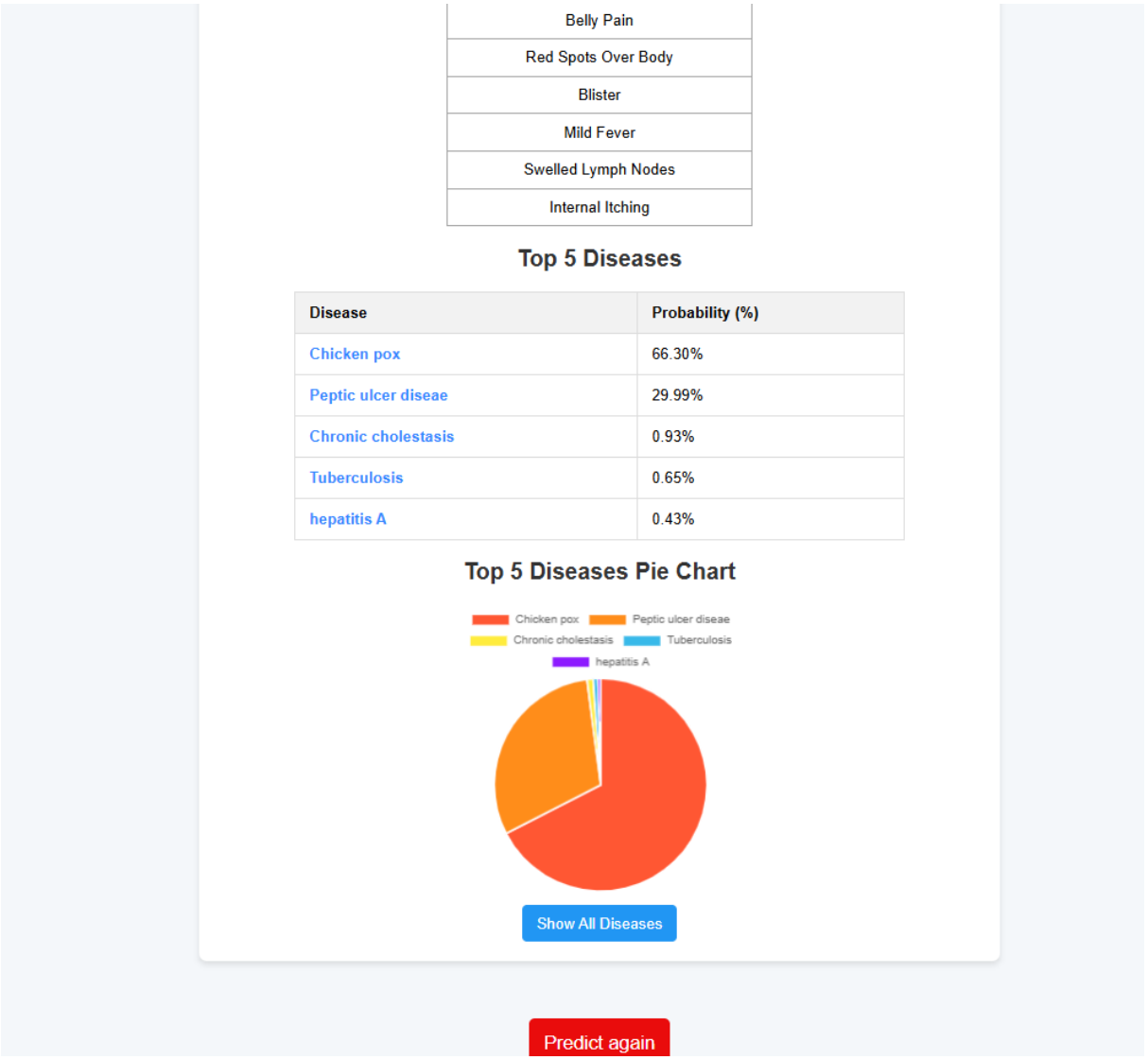
Klijentska aplikacija napravljena uz seminarski rad izrađena je u Pythonovom *frameworku* – Django. Za prikaz podataka u korisničkom sučelju korišten je HTML, standardni jezik za strukturiranje web-stranica, dok je CSS upotrijebljen za oblikovanje elemenata i vizualnu prilagodbu dizajna. JavaScript, kao dinamički skriptni jezik, omogućio je interaktivnost aplikacije, poboljšavajući korisničko iskustvo.

Pristupom na web aplikaciju *Illness Predictor*, prikazana je početna stranica koja korisnika usmjerava na načine korištenja web aplikacije. Ispod toga, vidljive su grupe simptoma, kao što su kardiovaskularne bolesti, bolesti kože i slično, a klikom na svaku od grupa simptoma, prikaz se proširi prikazujući popis svih simptoma vezanih uz tu grupu. Simptome je moguće označiti, a nakon unosa svih željenih simptoma, za predviđanje bolesti potrebno je kliknuti na gumb *Submit*. Zbog veće preciznosti i rizika koji donosi netočno predviđanje bolesti iako se koristi samo u savjetodavne svrhe ljudima i ne mijenja stručno mišljenje liječnika, nije dopušteno predviđanje bolesti na manje od tri unesena simptoma. Početna stranica s prikazom unosa simptoma prikazana je na slici 13.



Slika 13. *Prikaz stranice za unos simptoma*

Nakon pritiska na gumb *Submit*, model obrađuje korisničke simptome te generira listu unesenih simptoma koje je korisnik unio te za njih predviđene bolesti u sortiranom redoslijedu, od one najvjerojatnije do manje vjerojatnih. Zbog većeg broja bolesti koje se predviđaju, početno je prikazano pet najvjerojatnijih, a moguće je proširiti i izlistati sve bolesti s njihovim vjerojatnostima. Za pet bolesti s najvećom vjerojatnosti, izrađen je i tortni grafikon (engl. *pie chart*) koji vizualno prikazuje koje su bolesti najvjerojatnije, bez potrebe za detaljnijim uvidima u listu bolesti. Primjer prikaza stranice za predviđene bolesti, prikazan je na slici 14.



Slika 14. *Pregled stranice za predviđene bolesti*

Nakon predviđenih bolesti, ako korisnik nije zadovoljan unosom, može ga ponoviti pritiskom na gumb *Predict again*, a ako je zadovoljan te želi saznati više o nekoj o predviđenih bolesti, može pritisnuti na hipervezu te bolesti, odnosno na naziv bolesti u tablici. Pritiskom na hipervezu, korisnika se usmjerava na službenu stranicu *Mayo Clinic*, gdje je već unesena

odabrana bolest, te korisnik može istražiti više o toj bolesti, što uključuje rizične faktore, popis uobičajenih simptoma, savjete za liječenje i slično.

3.4. Dodatno

3.4.1. Izlaganje API-ja za dohvaćanje podataka iz baze podataka

Kako bi aplikacija bila još praktičnija, kreirana je i baza podataka koja omogućava spremanje podataka o korisnicima i korisničkim predikcijama. Baza podataka kreirana je uz pomoć PostgreSQL-a, koji pruža pouzdanu platformu za pohranu i upravljanje podacima unutar baze. S obzirom na to da je u aplikaciju uvedena baza podataka, omogućena je registracija i prijava korisnika u aplikaciju, što je i uvjet za daljnje korištenje iste. Podaci o korisnicima obuhvaćaju informacije kao što su e-mail, korisničko ime i zaporka, time omogućujući autentikaciju korisnika unutar aplikacije. Zapisi o predikcijama bolesti (*IllnessPrediction*) unutar baze podataka povezan je s određenim korisnikom putem stranog ključa (engl. *foreign key*), a također postoje i polja *symptoms* i *predictions* koji koriste JSON format za pohranu simptoma koje je korisnik unio i predikcija koje je generirao model. Model predviđanja bolesti prikazan je na slici 15.

```python
class IllnessPrediction(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, null=True, blank=True)
    symptoms = models.JSONField()
    predictions = models.JSONField()
```

Slika 15. *Prikaz modela IllnessPrediction*

Kako bi se omogućio lakši pristup podacima iz baze podataka, razvijen je API koristeći Django REST Framework. Ovaj API omogućava dohvaćanje podataka o korisnicima, simptomima, bolestima i predikcijama modela strojnog učenja putem RESTful GET *endpointova*. Implementacija ovog API-ja omogućuje fleksibilnu komunikaciju *frontend* i *backend* dijela aplikacije, kao i sigurnu integraciju s vanjskim sustavima.

Implementirani API sadrži sedam GET *endpointova* koji omogućuju:

1. Dohvaćanje podataka o pacijentu na temelju bolesti

```
GET /patients/diseases/?disease=diabetes%20
```

2. Dohvaćanje podataka o pacijentu na temelju simptoma

```
GET /patients/symptoms/?muscle_pain=false&muscle_weakness=false&obesity=true
```

3. Dohvaćanje podataka o bolesti na temelju simptoma

```
GET /diseases/?coma=false&irritability=false&obesity=true
```

4. Dohvaćanje podataka o bolesti na temelju ID-a predikcije

```
GET /diseases/prediction/?id=31
```

5. Dohvaćanje podataka o simptomima na temelju bolesti

```
GET /symptoms/?disease=psoriasis
```

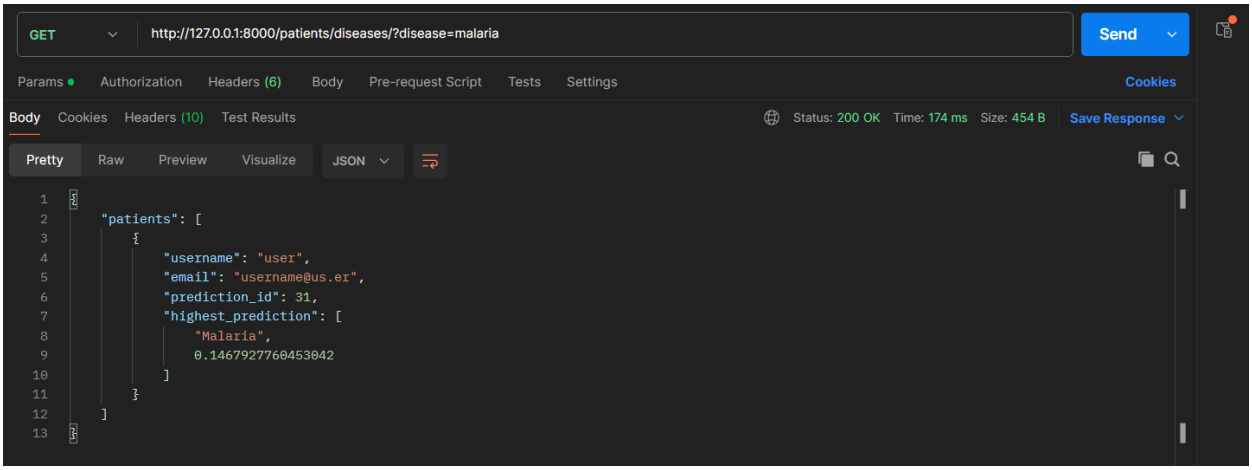
6. Dohvaćanje podataka o simptomima na temelju ID-a predikcije

```
GET /symptoms/prediction/?id=28
```

7. Dohvaćanje podataka o predikciji na temelju ID-a predikcije

```
GET /predict/32/
```

U nastavku je prikazan rad API-ja za dohvaćanje podataka o korisniku na temelju bolesti. Na slici 16. prikazan je primjer upotrebe, gdje se na temelju unesenih bolesti *Chicken pox* i *Malaria* dohvaćaju podaci o korisnicima koji su u predviđanju imali neku od tih bolesti kao najvjerojatniju, te su osim korisnika dohvaćeni i podaci o tome koja je točno bolest u pitanju i *prediction_id* koji omogućuje daljnji uvid u predviđanje. Za testiranje API-ja korišten je Postman, alat koji omogućava jednostavno slanje HTTP zahtjeva i pregled odgovora od servera. Postman je korišten kako bi se provjerilo ispravno funkcioniranje svih GET *endpointova* razvijenih unutar aplikacije.



Slika 16. *Prikaz zahtjeva i odgovora API-ja u JSON formatu za dohvaćanje pacijenata na temelju bolesti*

3.4.2. Podešavanje hiperparametara modela

Za optimizaciju modela strojnog učenja korišten je Azure ML Designer, unutar kojeg su hiperparametri konfigurirani uz pomoć modula *Tune Model Hyperparameters*, koji je korišten za svaki netrenirani model. Hiperparametri su uspoređivani s obzirom na metriku *Accuracy*. Osim podešavanja hiperparametara, broj značajki je prije podjele podataka na trening i testni skup povećan na 87, čime se pokušala povećati točnost predviđanja modela. Na slici 17. prikazan je skup uspoređivanih parametara korištenih tijekom podešavanja hiperparametara svih modela.

LOGISTIC REGRESSION

Create trainer mode	ParameterRange
Optimization tolerance	0.001; 0.00001; 0.000001; 0.0000001
L2 regularization weight	0.001; 0.01; 0.1; 1.0; 10
Random number seed	

BOOSTED DECISION TREE

Create trainer mode	ParameterRange
Maximum number of leaves per tree	20; 30; 50; 100; 150
Minimum number of samples per leaf node	1; 5; 10; 20; 50
Learning rate	0.025; 0.05; 0.1; 0.2; 0.4
Number of trees constructed	100; 150; 200; 500; 1000
Random number seed	

SUPPORT VECTOR MACHINE (OvA)

Create trainer mode	ParameterRange
Number of iterations	10; 50; 100; 500; 1000; 2000; 5000;
Lambda	0.00001; 0.0001; 0.001; 0.01; 0.1
Normalize features	True
Random number seed	

NEURAL NETWORK

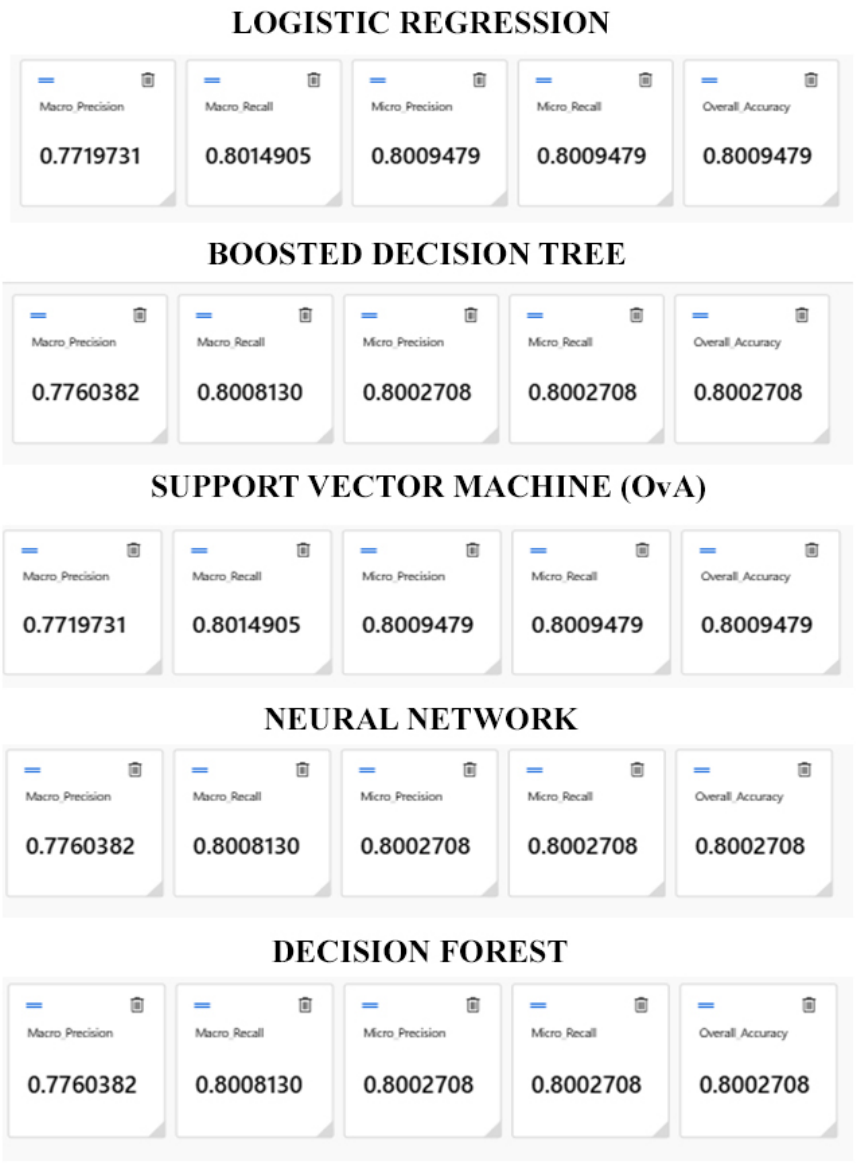
Create trainer mode	ParameterRange
Hidden layer specification	Fully-connected case
Number of hidden nodes	87,87
Learning rate	0.1; 0.01; 0.001
Number of learning iterations	100; 300; 500
The momentum	0.9
Shuffle examples	True
Random number seed	

DECISION FOREST

Create trainer mode	ParameterRange
Number of decision trees	8; 16; 32; 64; 128
Maximum depth of the decision trees	32; 32; 64; 128
Minimum number of samples per leaf node	1; 2; 5; 10
Resampling method	Bagging Resampling

Slika 17. *Korišteni hiperparametri za podešavanje*

Nakon usporedbe prethodno prikazanih hiperparametara i treniranja modela s tim hiperparametrima, evaluacijom modela čiji su rezultati prikazani na slici 18., utvrđeno je da su *Support Vector Machine (OvA)* i *Logistic Regression* imali najveću ukupno točnost od 80,95%, dok su ostala tri modela imali 80,03%. Za stvaranje *Real-time inference pipelinea* odabran je *Support Vector Machine* model te je pušten u produkciju. S obzirom na to da je promijenjen broj značajki, promijenjena je forma za unos simptoma unutar aplikacije.



Slika 18. *Metrike modela nakon podešavanja hiperparametara*

Novi model pokazao je značajne razlike u predviđanju bolesti. Prije optimizacije hiperparametara modela korišten je *Neural network*, koji je pravio jasne razlike u vjerojatnostima pojavljivanja određene bolesti, te su vrlo često vjerojatnosti bolesti koje ne odgovaraju simptomima iznosile 0%. Korištenjem *Support Vector Machinea*, razlike u vjerojatnostima između bolesti znatno su manje, pa je i najvjerojatnija bolest u predviđanju relativno niske vjerojatnosti.

Razlike u predviđanjima između *Neural Networka* i *Support Vector Machine* modela proizlaze iz njihovih temeljnih principa rada. *Neural network* model uči složene odnose između simptoma i bolesti, što mu omogućava da s visokom sigurnošću dodijeli visoku vjerojatnost najizglednijoj dijagnozi, dok ostale bolesti često dobiju vjerojatnost blisku nuli. *Support Vector Machine* koristi drugačiji pristup klasifikaciji, pri čemu pokušava pronaći optimalne granice između klasa. Budući da je dizajniran za binarnu klasifikaciju, kod višeklasne klasifikaciječesto daje ujednačenije vjerojatnosti za različite klase, bez jasnog favoriziranja jedne.

Neural network model korisniji je kada je važno jasno razlikovati najvjerojatniju bolest od svih ostalih, dok *Support Vector Machine* pruža širi spektar mogućih bolesti, što može biti korisno u situacijama gdje je potrebno provjeriti i druge parametre koji ne uključuju simptome pri donošenju odluka o tome koja je bolest stvarno u pitanju.

4. ZAKLJUČAK

Za stvaranje web aplikacije za predviđanje bolesti na temelju unesenih simptoma potrebno je pronaći dobar podatkovni skup. Podatkovni skupovi vezani uz medicinske dijagnoza uglavnom se rijetko mogu pronaći na internetu, zbog očuvanja podataka pacijenata. Nakon odabira podatkovnog skupa, obično je potrebno izvršiti određene transformacije, kao što su normalizacija skupa podataka, redukcija dimenzionalnosti skupa, rukovanje nedostajućim vrijednostima i slično. Nakon predobrade podataka, ovisno o kvaliteti i kvantiteti podatkovnog skupa, u određenom se omjeru dijeli na dva podskupa: trening skup i testni skup. Trening skup služi za treniranje modela strojnog učenja, koji u konkretnom slučaju zadatka ovog seminarskog rada, na temelju prisutnosti određenih simptoma predviđa bolest, a testni skup testira sposobnost izrađenog modela da predvidi pripremljene podatke koji su mu do tada neviđeni. Model koji se koristi u konačnoj web aplikaciji – *Multiclass Neural Network*, odabran je usporedbom između više modela s obzirom na vrijednosti njihovih metrika. Nakon što je model stvoren, potrebno je stvoriti *Real-time inference pipeline*, kojemu se nakon puštanja u produkciju može pristupiti putem API-ja. Nakon toga, izrađena je aplikacija unutar okvira Django, koja omogućava korisniku odabir simptoma te ispis vjerojatnosti bolesti koje je generirao model.

Izazovi koji su se pojavljivali u izradi konačne web aplikacije najviše su se ticali podatkovnog skupa. Iako mnoštvo podatkovnih uzoraka, podatkovni skup sadrži 41 bolest koju treba predvidjeti na temelju 132 simptoma. Predobrada podataka predstavljala je najveći problem, jer je bez smanjenja dimenzionalnosti dolazilo do prenaučenosti svih modela, odnosno, na trening podacima točnost je bila iznimno visoka (što je i poželjno kod medicinsko-orijentiranih modela), međutim na testnim podacima zbog „prokletstva dimenzionalnosti“ predviđanje je bilo izrazito loše. S druge strane, smanjenjem dimenzionalnosti dolazi se u opasnost ignoriranja značajki koje su jako vezane uz određenu bolest, pa predviđanje tih bolesti od strane modela, iako se u stvarnosti možda pojavljuju i češće od drugih, rezultira niskom vjerojatnosti i teško je ugoditi modelu da ju s velikom vjerojatnošću pogodi.

U konačnici, unatoč izazovima vezanim uz podatke i modeliranje, razvijena web aplikacija predstavlja vrijednu demonstraciju primjene strojnog učenja u zdravstvenom sektoru. Daljnjim radom, ovakve aplikacije mogle bi postati neprocjenjiv alat u pružanju podrške pacijentima i liječnicima.

5. POVEZNICE I LITERATURA

Programskom je rješenju moguće pristupiti preko:

Programsko rješenje na GitHubu
ML model
Web rješenje

[1] „Microsoft Learn: Multiclass Logistic Regression component“ [online], dostupno na: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/multiclass-logistic-regression?view=azureml-api-2 [26. 2. 2025.]

[2] „Microsoft Learn: Multiclass Boosted Decision Tree“ [online], dostupno na: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/multiclass-boosted-decision-tree?view=azureml-api-2 [26. 2. 2025.]

[3] „Microsoft Learn: Multiclass Neural Network component“ [online], dostupno na: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/multiclass-neural-network?view=azureml-api-2 [26. 2. 2025.]

[4] „Microsoft Learn: Multiclass Decision Forest component“ [online], dostupno na: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/multiclass-decision-forest?view=azureml-api-2 [26. 2. 2025.]

[5] „Microsoft Learn: Two-Class Support Vector Machine component“ [online], dostupno na: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/two-class-support-vector-machine?view=azureml-api-2 [26. 2. 2025.]

[6] „Microsoft Learn: One-vs-All Multiclass“ [online], dostupno na: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/one-vs-all-multiclass?view=azureml-api-2 [26. 2. 2025.]